

## Description

# [CIRCUIT AREA MINIMIZATION USING SCALING]

### BACKGROUND OF INVENTION

[0001] Technical Field

[0002] The present invention relates generally to circuit design, and more particularly, to a method, system and program product for scaling and area minimization of a circuit design.

[0003] Related Art

[0004] Circuit compaction is an integral part of integrated circuit (IC) chip design methodology for design optimization or migration. In conventional circuit compaction, a circuit design is reduced to its minimum legal size by building a constraint graph that represents the shape edges as nodes, and ground rules and topological relationships as arcs. Each node value represents the edge's current position, and each arc value represents the minimum separa-

tion between the edges connected by the arc. In order to illustrate, FIG. 1 shows an illustrative circuit design 8 constraint graph. In a typical circuit compactor, reduction of circuit design 8 is completed by assigning a graph node 10a-d to represent a location of every shape edge 12 that is perpendicular to the direction of compaction (CD), and one graph node 10e-f each to a "source" 14 and a "sink" 16, which are the boundaries of the design. In the example shown in FIG. 1, two rectangles 18 are positioned side-by-side. Accordingly, if compaction occurs in an "X" direction, a graph node 10a-f would represent every vertical edge 12 of each rectangle 18. Each graph node 10a-f has a value (its X position).

[0005] In FIG. 1, the locations to be assigned are X1, X2, X3, X4 and sink 16. Source 14 is fixed at the origin. In addition, each graph node 10a-f has an arc 20a-f that connects it to its neighbors, which represent constraints (ground rule values) on the locations X1-X4 and sink 16. For example, arc 20a and 20e may be half the minimum spacing for wires on metal layer one (M1) and may be, for example, 0.1 microns (these arcs thus represent a boundary condition); arc 20c may be the minimum M1 spacing, e.g., 0.2 microns; and arcs 20b and 20d may be the minimum

width for an M1 wire, e.g., 0.2 microns. Arc 20f will be ignored for now.

[0006] In order to compact the design to a minimum size, a graph longest path algorithm is used to find the minimum distance from the boundary to each node, subject to the restrictions represented by the arcs. That is, classical compaction will assign the locations X1, X2, X3, X4 and sink 16 using the "longest path" algorithm. The longest path algorithm will assign, for example:

[0007]	X1	Source + 20a	$0+0.1$	0.1
	X2	X1+20b	$0.1+0.2$	0.3
	X3	X2+20c	$0.3+0.2$	0.5
	X4	X3+20d	$0.5+0.2$	0.7
	Sink	X4+20e	$0.7+0.1$	0.8

[0008] This configuration compacts rectangles 18 to their minimum legal size in the direction of compaction (CD).

[0009] One problem with conventional longest-path based compaction is that it is incapable of meeting all of the constraints which users place on the design. A completed circuit design is subject to constraints in the form of ground rules, which are well understood by conventional compaction. However, there are a number of other constraints that the designer has in mind while laying out the circuit.

These constraints may relate to issues such as performance or yield objectives. Many of these constraints may not be able to be described geometrically, making them difficult to describe to a conventional compaction tool. For constraints that can be described, entering all of the user constraints into a compaction tool can be a task that is nearly as difficult as re-implementing the circuit from scratch.

[0010] Conventional longest path compaction algorithms cannot compact a design that contains a "positive cycle." A positive cycle typically results if there are illegal shape configurations (ground rule incorrect starting point), or if there are constraints that restrict the area of the layout to less than the required size. If there is a positive cycle, then there is no longest path. Removing the positive cycle in a typical compactor does not correct the illegality that brought about the cycle, e.g., a broken ground rule or unachievable constraint. Relative to FIG. 1, a positive cycle can be illustrated by assuming a boundary constraint that fixes arc 20f at 0.6. In this case, every time a conventional compactor traverses the path 20a, 20b, 20c, 20d, 20e, 20f, the total cost increases by 0.2 (i.e.,  $0.1 + 0.2 + 0.2 + 0.2 + 0.1 - 0.6$ ). In a shortest path algorithm, the equiva-

lent problem is referred to as a "negative cycle." In any case, the positive cycle represents a conflict in the constraints (arcs 20a–20f), not all of which can be met.

[0011] Another problem with classical compaction approaches is created by how the techniques create the maximum changes possible to the design. In particular, conventional compaction algorithms conduct "plowing" or squeezing of every shape as far as possible in the compaction direction, which can destroy symmetry or alignment that was present in the original design. For example, FIGS. 2A–2B illustrate two rows of elements in which the top row contains six (6) elements 30a–30f, and the bottom row contains only three (3) elements 32a–32c. In addition, every other top row element 30a, 30c, 30e is connected to one of the bottom row elements 30a, 30b, 30c, respectively. FIG. 2A illustrates an un-compacted layout, FIG. 2B illustrates a compacted layout using conventional compaction techniques including a "jog insertion" implementation such that wires can be bent as required. As illustrated in FIG. 2B, the layout is compacted in a legal manner, but the relative positions of elements 30a–30f and 32a–32c do not encompass the designer's original intent. This plowing of the layout prevents many designers from using a com-

pactor because of the unpredictable and inelegant outcomes to the layout, which in turn prevents widespread use of compaction technology.

[0012] Minimum perturbation analysis (hereinafter referred to as "minpert") is used to implement ground rule adjustments in circuit designs in a way that makes the minimum number of changes necessary. In particular, when a design has a minpert analysis applied, any of the arcs 20a–20f (FIG. 1) that are legal, are frozen. Any arcs 20a–20f that are not legal are fed to a linear system solver, which attempts to fix every arc that does not have a currently legal value, and minimize the overall differences in positions, i.e.,  $X_2 - X_1 \geq 0.2$ . If  $X_2 - X_1$  is currently 0.1, then the system will try to change  $X_2 - X_1$ , while minimizing  $X_2(\text{new}) - X_2(\text{old})$ , and minimizing  $X_1(\text{new}) - X_1(\text{old})$ . U.S. Patent No. 6,189,132 to Heng et al., which is hereby incorporated by reference, discloses a design rule correction system and method that implements minpert analysis. Unfortunately, minpert analysis does not address area minimization.

[0013] Rudimentary scaling has been used to perform compaction, but has led to unacceptable results for widespread use.

[0014] In view of the foregoing, there is a need in the art for an

improved compaction method, system and program product for compacting a circuit design.

## **SUMMARY OF INVENTION**

[0015] The present invention provides a method, system and program product that implements area minimization of a circuit design while respecting the explicit and implicit design constraints, in the form of ground rules and user intent. A longest path algorithm is used to generate a scaling factor. The scaling factor is used to reduce the size of the circuit design to the minimum legal size. The scaling may be followed by application of minpert analysis to correct any errors introduced by the scaling. The resulting design is shrunk (or expanded) with all elements shrinking (or growing) together by the same factor, and with the relative relationships of elements maintained. In addition, the invention is operational in the presence of a positive cycle, can be run with scaling that freezes the sizes of any structure or ground rule, and can be applied to technology migration.

[0016] A first aspect of the invention is directed to a method for minimizing area of a circuit design, the method comprising the steps of: applying a longest path algorithm to the circuit design to determine a minimum legal size; calcu-

lating a scaling factor based on a ratio of the minimum legal size divided by a pre-compaction size; and scaling the circuit design using the scaling factor while maintaining an objective.

[0017] A second aspect of the invention is directed to a system for minimizing area of a circuit design, the system comprising the steps of: means for applying a longest path algorithm to the circuit design to determine a minimum legal size; means for calculating a scaling factor based on a ratio of the minimum legal size divided by a pre-compaction size; and means for scaling the circuit design using the scaling factor while maintaining an objective.

[0018] A third aspect of the invention is directed to a computer program product comprising a computer useable medium having computer readable program code embodied therein for minimizing area of a circuit design, the program product comprising: program code configured to apply a longest path algorithm to the circuit design to determine a minimum legal size from a boundary to an opposite boundary; program code configured to calculate a scaling factor based on a ratio of the minimum legal size divided by a pre-compaction size from the boundary to the opposite boundary; program code configured to scale



the circuit design using the scaling factor; and program code configured to apply a minimum perturbation analysis to the scaled circuit design.

[0019] The foregoing and other features of the invention will be apparent from the following more particular description of embodiments of the invention.

#### **BRIEF DESCRIPTION OF DRAWINGS**

[0020] The embodiments of this invention will be described in detail, with reference to the following figures, wherein like designations denote like elements, and wherein:

[0021] FIG. 1 shows an illustrative prior art constraint graph for use with a prior art compaction algorithm.

[0022] FIG. 2A shows another illustrative circuit design.

[0023] FIG. 2B shows application of a prior art compaction algorithm to the illustrative circuit design of FIG. 2A.

[0024] FIG. 3 shows a block diagram of a minimization system according to the invention.

[0025] FIG. 4 shows a flow diagram of operation methodology of the minimization system of FIG. 3.

#### **DETAILED DESCRIPTION**

[0026] With reference to the accompanying drawings, FIG. 3 is a block diagram of a circuit design area minimization sys-

tem 100 (hereinafter "minimization system") in accordance with the invention. Minimization system 100 includes a memory 112, a central processing unit (CPU) 114, input/output devices (I/O) 116 and a bus 118. A database 120 may also be provided for storage of data relative to processing tasks. Memory 112 includes a program product 122 that, when executed by CPU 114, comprises various functional capabilities described in further detail below. Memory 112 (and database 120) may comprise any known type of data storage system and/or transmission media, including magnetic media, optical media, random access memory (RAM), read only memory (ROM), a data object, etc. Moreover, memory 112 (and database 120) may reside at a single physical location comprising one or more types of data storage, or be distributed across a plurality of physical systems. CPU 114 may likewise comprise a single processing unit, or a plurality of processing units distributed across one or more locations. I/O 116 may comprise any known type of input/output device including a network system, modem, keyboard, mouse, scanner, voice recognition system, CRT, printer, disc drives, etc. Additional components, such as cache memory, communication systems, system software,

etc., may also be incorporated into system 100.

[0027] As shown in FIG. 3, minimization system 100 may interact or be configured to operate as part of a larger circuit design system 102. Design system 102 provides a circuit design 90 to be compacted, and receives a compacted circuit design 92. Program product 122 may include a scaling factor determinator 124 including a longest path algorithm applicator 126 and a scaling factor calculator 128; a scaler 130 including a minimum perturbation analyzer 132; and an element size fixer 134 and other system components 136. Other system components 136 may include any other hardware or software necessary to achieve minimization system 100 functioning not distinctly described herein.

[0028] Turning to FIG. 4, in conjunction with FIG. 3, operation methodology for minimization system 100 will now be described. As an alternative precursor step S1, element size fixer 136 may be implemented to denote one or more elements 94 of circuit design 90 to be fixed in size. For example, size of a gate or other device, or a wire may be fixed prior to a longest path algorithm applying step and a scaling step, as will be described below. In one particular embodiment, a gate in circuit design 90 is fixed in

size.

[0029] In a first step S2, a scaling factor is determined by scaling factor determinator 124. In one embodiment, step S2 includes, at step S2A, applicator 126 applying a longest path algorithm to circuit design 90 having a pre-compaction size (Z1) to determine a minimum legal size (denoted Z2) from a boundary (i.e., source) 96 to an opposite boundary (i.e., sink) 98. Since various longest path algorithms are well known in the industry, details of their operability will not be provided. Generally, however, positions are assigned to the vertices that minimize the distance from source 96 to sink 98, as shown in FIG. 3, which minimizes the design width in a direction of compaction. In one embodiment, the longest path algorithm is applied with assigning of a source 96 and a sink 98 only. In addition, where the longest path algorithm cannot be completed due to a "positive cycle," applicator 124 ignores or makes an assumption regarding selected design constraints, i.e., backward arcs.

[0030] A "positive cycle" is a design rule constraint or error, which prevents a classic longest path algorithm from finding a longest path through the circuit. A positive cycle typically results if there are illegal shape configurations

(e.g., ground rule incorrect starting point), or if there are constraints that restrict the area of circuit design 90 to less than the required size. If there is a positive cycle, then the longest path will not exist. Scaling factor determinator 124 identifies the constraints that cause the positive cycle, and proceed with an approximate solution. Alternatively, the invention may detect the positive cycle, remove one or more constraints that create the cycle, and determine a new minimum legal size. This new minimum 'legal' size will be smaller than the actual legal minimum for the circuit. However, the minimum perturbation analysis, as will be described below, will correct for these illegalities. Accordingly, the invention is not sensitive to the positive cycle phenomenon.

[0031] Returning to FIG. 4, in step S2B, a scaling factor is calculated by calculator 128 based on a ratio of minimum legal size (Z2) divided by pre-compaction size (Z1) from boundary 96 to opposite boundary 98. In calculating a scaling factor using this step, the present invention eliminates the hit-or-miss approximations of conventional scaling systems.

[0032] In step S3, circuit design 90 is scaled. In particular, at step S3A, circuit design 90 is scaled by scaler 130 using scal-

ing factor ( $Z2/Z1$ ). As used herein, "scaling" means taking circuit design 90 and multiplying the coordinates of every vertex by scaling factor ( $Z2/Z1$ ). For example, an 80% scaling would involve multiplying every vertex's X and Y coordinates by 0.80, which has the effect of shrinking or expanding the design uniformly. As noted above, element(s) 94 sizes may be fixed as needed (optionally not scale actual devices or wires). In addition, scaler 130 can also scale elements (e.g., gates, other devices, wires, etc.) as a group or individually, thus re-sizing devices on-the-fly. Scaler 130 can also run various methods of scaling wires.

[0033] The above-described scaling may result in some illegalities. Accordingly, in step S3B, as an alternative step, a minimum perturbation (minpert) analysis may be conducted on compacted circuit design 92 by minpert analyzer 132. This step corrects any ground rule error(s) in compacted circuit design 92 by making the minimum number of changes necessary. Implementation of minpert analysis may be similar to that disclosed in U.S. Patent No. 6,189,132 to Heng et al. Since the original scaling factor ( $Z2/Z1$ ) was legal, the boundary of compacted circuit design 92 should not grow during this step.

[0034] As shown in compacted circuit design 94 (FIG. 3), the above-described invention can compact a circuit design 90 while maintaining objectives such as: a distance between elements, alignment of elements, a size of an element, relative positions of elements, and overall pattern of the un-compacted circuit design. The design is not "plowed" in one direction or the other, which previously may have prevented designers from using a compactor because of the unpredictable and inelegant outcomes to the layout. Accordingly, the invention generates a compacted circuit design that looks like the original, but is smaller. Another advantage of the present invention is that it can be applied to migration.

[0035] In the previous discussion, it will be understood that the method steps discussed are performed by a processor, such as CPU 114 of system 100, executing instructions of program product 122 stored in memory. It is understood that the various devices, modules, mechanisms and systems described herein may be realized in hardware, software, or a combination of hardware and software, and may be compartmentalized other than as shown. They may be implemented by any type of computer system or other apparatus adapted for carrying out the methods de-

scribed herein. A typical combination of hardware and software could be a general-purpose computer system with a computer program that, when loaded and executed, controls the computer system such that it carries out the methods described herein. Alternatively, a specific use computer, containing specialized hardware for carrying out one or more of the functional tasks of the invention could be utilized. The present invention can also be embedded in a computer program product, which comprises all the features enabling the implementation of the methods and functions described herein, and which – when loaded in a computer system – is able to carry out these methods and functions. Computer program, software program, program, program product, or software, in the present context mean any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after the following: (a) conversion to another language, code or notation; and/or (b) reproduction in a different material form.

[0036] While this invention has been described in conjunction with the specific embodiments outlined above, it is evi-



dent that many alternatives, modifications and variations will be apparent to those skilled in the art. Accordingly, the embodiments of the invention as set forth above are intended to be illustrative, not limiting. Various changes may be made without departing from the spirit and scope of the invention as defined in the following claims.